

**DEVICE STATUS MONITORING SYSTEM, DEVICE STATUS
MONITORING METHOD, AND A DATA STORAGE MEDIUM AND
OBJECT PROGRAM THEREFOR**

5 **BACKGROUND OF THE INVENTION**

1. Field of the Invention

The present invention relates generally to a device status monitoring system, a device status monitoring method, and a data storage medium and object program recording the same. More particularly, the invention relates to a
10 device status monitoring system and method that use an OPOS service object to continuously record status change data indicating a change in the status of a device in a POS terminal system to a status change recording unit based on specified recording conditions, and a data storage medium and object program recording the same.

15

2. Description of Related Art

Multifunction POS (point-of-sale) systems can be assembled without specifically considering the makes and models of devices used to configure the POS system by using a standard application programming interface (API) to
20 develop a software application providing POS system functions. OPOS (OLE for Retail POS) is an international standard defining a standardized interface for handling devices (such as printers and displays) used in the POS terminals (referred to below as POS terminal systems) of a POS system. An OPOS standard object uses OLE (Object Linking and Embedding) controls, which are
25 supported by the Windows (a registered trademark of Microsoft Corp.) operating system (OS) running on the host devices of the POS system, to provide application developers with a set of API functions that make application development easier. Note that OPOS standard objects are referred to below as simply "OPOS."

Information needed to use OPOS, that is, information about all of the devices in the POS terminal system and settings for all of the devices (referred to below as the OPOS device environment configuration), is stored in a database (referred to below as the registry) of system configuration information managed by the Windows OS. This registry is a database for centrally managing all settings related to computer operation, including all device driver settings and application settings. Information about each device in the POS terminal system, including a device identifier and various settings for each device, stored in the registry as part of the OPOS device environment configuration settings is referred to below as the device information or the device profile. In other words, the OPOS device environment configuration settings include a device profile for each device in the system configuration.

Data indicating a change in the status of a device (referred to below as "status change data") that is reported to the application providing POS terminal system functions through OPOS is conventionally only part of the status change data detected from status data received from a particular peripheral device by the host computer running the application. More specifically, OPOS determines which of the device status change data must be reported to the application, and sends only the necessary status changes to the application. The problem with this is that the application is therefore unable to know all of the device status changes, which makes it difficult to troubleshoot errors and problems in the system or a particular device.

Devices can also use LEDs or other type of display to inform the POS terminal system user or developer, for example, that an error has occurred and even indicate the type of error. In a POS terminal system in which the device and host computer are integrated into a single unit, however, the device status (error) indicator is hidden inside the POS terminal system, and the POS terminal system user or developer cannot easily know what the problem reported by the device is. The device installation (where and how the device is installed) may also make the status display unreadable. The status display may even be

eliminated in order to reduce device cost. Because problems with a device cannot be known in real time in these cases, errors cannot be addressed. This means the source of the problem cannot be determined, and it takes longer to resolve problems and restore the device to normal working condition.

5

SUMMARY OF THE INVENTION

This invention is therefore directed to a solution for these problems, and an object of this invention is to provide a device status monitoring system and monitoring method using an OPOS service object to continuously record status change data indicating a change in the status of a device in a POS terminal system to a status change recording unit based on specified recording conditions.

We have researched solutions for the prior art problems described above. As a result we discovered that all changes in device status can be recorded by providing, in an OPOS service object for controlling a device connected to a host computer through the operating system of the host computer in a POS terminal system, a function for recording status change data, which is indicative of a change in device status, to a file or other data recording unit.

The invention we derived from this research is described briefly below.

A device status monitoring system according to the present invention has a status change data recording unit for sequentially recording, by means of a second object of a device control system, status change data indicating a change in a particular device status to a status change recording unit, also referred to below as a log file. The device control system is a system for controlling a specific device connected to the host computer of a POS terminal system through the operating system of the host computer. The device control system has a first object providing a device class interface to an application run under the operating system and capable of controlling the device. The second object of the device control system provides an interface for a specific device to the first object.

The status change data recording unit preferably has a receiving unit for receiving status data from the specific device, and a status change data detection

unit for detecting status data that changed to obtain the status change data based on the status data received by the receiving unit and the previously received status data.

Further preferably, the status change data recording unit has a recorded
5 data editor for editing the status change data to generate the status change recording data (also referred to below as log data), which is recorded to the log file.

The recorded data editor preferably has a recorded data evaluation unit
10 for determining, based on predefined recording conditions, if the status change data detected by the status change data detection unit is data to be recorded in the status change data log file.

The status change data includes error status and/or off-line status data, and the recording conditions include information indicating if the error status and/or off-line status data should be recorded. In addition to the type of status
15 change data to be recorded, the recording conditions can also specify the devices for which status change data is to be recorded, the name of the log file to which the status change data is recorded, and the maximum size of the log file.

Preferably, the recorded data editor has a recorded data generating unit
20 for converting the status change data detected by the status change data detection unit to a text message and adding time information to the text message to generate the log data.

In another aspect, the present invention provides a device status monitoring method having the same operation and effect as the monitoring system. In yet another aspect, the present invention provides a computer-
25 executable program that implements the device status monitoring method. The program can be recorded to and distributed using a data storage medium such as a Compact Disc (CD), floppy disk, hard disk, magneto-optical disc, Digital Versatile/Video Disc (DVD), magnetic tape, or semiconductor memory, or transmitted by a carrier wave over a communication channel.

Other objects and attainments together with a fuller understanding of the invention will become apparent and appreciated by referring to the following description and claims taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a typical POS terminal system configuration;

Fig. 2 is a block diagram of a POS terminal system;

Fig. 3 shows the relative position and function of OPOS in a POS terminal system;

Fig. 4 is a function block diagram of a device status monitoring system;

Fig. 5 is a flow chart of the device status monitoring process;

Fig. 6 is a flow chart of the status change data recording routine in the device status monitoring process;

Fig. 7 is a flow chart of the process for setting the recording conditions;

Fig. 8 shows an example of the status change data log written to the status change recording unit (log file);

Fig. 9 shows an example of a screen for setting the recording conditions for a specific device; and

Fig. 10A shows the device information in the OPOS device environment configuration settings grouped by device class for display, and Fig. 10B shows the device information grouped by connection port.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Preferred embodiments of the present invention are described below with reference to the accompanying figures. It will be noted that the following embodiments are shown by way of description only and shall not limit the scope of the invention. It will be obvious to one of ordinary skill in the related art that various alternative embodiments can be achieved by replacing some or all of the elements described below with equivalent elements, and that all such variations are included in the scope of this invention.

Fig. 1 shows the configuration of a typical POS terminal system 100. This POS terminal system 100 has a host computer 101, printer 102 for printing receipts and a transaction journal, customer display 103 for displaying purchase information, and cash drawer 104 for holding money and checks. The host computer 101 stores a software application providing the various functions of the POS terminal system 100. The host computer 101 is connected by a communication bus 110 to a printer 102, the customer display 103, and the cash drawer 104. Various communication protocols can be used on the communication bus 110, including serial, parallel, and network (such as IEEE (Institute of Electrical and Electronic Engineers) 802), or USB (Universal Serial Bus). Although a POS terminal system is specifically described, the present invention may be broadly applied to other data processing systems that include peripheral devices connected to a host computer.

Fig. 2 is a block diagram of the host computer 101. The host computer 101 has an interface 201 for connecting to the communication bus 110, a disk drive 202 for reading software and information from a CD (Compact Disc), DVD (Digital Video/Versatile Disc) or other medium, a CPU 203, RAM 204 for storing the program run by the CPU 203 and used as working memory by the CPU 203, a hard disk 205 for storing the program and data used by the CPU 203, a monitor or other type of display device 206 for presenting messages and information for the user, a floppy disk drive 207 for reading software and other information from a floppy disk, a keyboard and mouse or other type of input device 208 enabling user input to the system, and a bus 210 connecting these various components.

Fig. 3 illustrates the role of OPOS in a POS terminal system. As shown in Fig. 3, OPOS is located between the OS and the application controlling the devices in the POS terminal system, and provides a standardized interface for connecting the host computer and peripheral devices based on defined specifications.

OPOS has two layers, the control object (CO) layer and service object (SO) layer. Control objects are provided for each device class, such as printers and displays, and control the device-application interface. Service objects are provided for each device, such as for each printer model. The service objects control device operation by way of the OS.

The application controls a particular device using methods and properties, and receives feedback about the result of device control from the controlled device using events and properties. OPOS converts process requests from the application to commands supported by the device and sends the commands to the addressed device, and receives the status indicating the process result from the device. It also receives status data (report) including information about the state(s) of the device. The service object receives all device status data. As indicated in Fig. 3, the status change data recording function is a part of the service object. Status change data is extracted by searching the status data received by the service object for a device status that changed since the previous status data was received. The status change data required by the application is selected and converted (edited) to a specific transmission format, and then passed to the control object. The control object uses events to notify the application of a status change when status change data is received from a service object.

Fig. 4 is a function block diagram of a device status monitoring system. As shown in Fig. 4, the host computer 101 comprises a controller 401, receiving unit 402, status change data recorder 403, and recording conditions input unit 404. The receiving unit 402 receives all status data sent from the device 420. The status change data recorder 403 compares the status data received by the receiving unit 402 with the previously received status data stored in status data memory 410. The status change data recorder 403 edits the status change data based on the status data that changed and the recording conditions stored in the recording conditions memory 412 to continuously update the log file 411. More specifically, a data log is generated and stored to a specific log file 411 (see Fig.

8) by converting status change data meeting the recording conditions to a text message, getting the time and date that the status data was received from a real-time clock IC of the host computer 101 by way of the OS, and writing the text message and time information as an entry in the log file. The received status data is also stored to the status data memory 410.

The recording conditions input unit 404 stores the recording conditions inputted from a keyboard, mouse, or other input device of the host computer (not shown in the figure) to the recording conditions memory 412.

The controller 401 cooperatively controls the receiving unit 402, status change data recorder 403, and recording conditions input unit 404. Part of controller 401 functionality, receiving unit 402, status change data recorder 403, and status data memory 410 are contained in a service object. In other words, the service object provides a status change data recording function for recording all status data received from a device.

Fig. 5 is a flow chart of the device status monitoring process. This process starts with the receiving unit 402 receiving status data from a device (S501). The status change data recorder 403 then performs a status change data recording routine (S502) which includes comparing the received status data with the previously received status data, editing the status change data based on the recording conditions stored in the recording conditions memory 412 and the status data for which a change was detected in step S501, recording the edited log entry in the log file 411, and then storing the received status data in the status data memory 410.

Whether there is another process to run is then determined based on the status change data (S503). If there is (S503 returns yes), that process is run (S504), and the procedure then loops back to step S501 and waits until status data is received for another device. If there is no other process to run (S503 returns no), the procedure loops back to step S501 and waits until status data is received for another device. These other processes include, for example, reporting

status change data to the application through a control object if the status change data is information that must be reported to the application.

Fig. 6 is a flow chart of the status change data recording process shown as subroutine S502 of the device status monitoring process shown in Fig. 5. The status change data recorder 403 first reads the required recording conditions from recording conditions memory 412 (S601). The required recording conditions indicate the device for which the status change data is to be recorded, the type of information to record, the filename to which the log is recorded, and the maximum size of the log file. The information types include error status and off-line status data, either or both of which can be selected as information to log. If neither is selected, the status change data recording function is inactive. Error status data includes information about errors that occurred and error recovery, such as that a mechanical error occurred and was corrected, or that an automatic paper cutter error occurred and was corrected. Off-line status data includes data indicating that the device went off-line and was then restored to an on-line status. This status data is generated, for example, when the printer goes off-line because the cover was opened, and when the printer comes back on-line because the cover was closed, or when the printer goes off-line because there is no paper and then returns on-line because paper was loaded.

The previously received status data is then read from the status data memory 410 (S602) and compared with the status data just received to detect any change in one or more status data (S603). Data indicating that a particular device status has changed is referred to herein as status change data. If a status data from a device is 1 byte long and each bit indicates a different device status, plural changes in device status could be detected from a single status data.

Whether the detected status change data should be recorded is then determined based on the recording conditions read from the recording conditions memory 412 (S604). This step is also referred to as a status change data evaluating step. More specifically, this step detects whether an error or off-line state was reported. If the status change data should be recorded (S604 returns

yes), the status change data log entry is generated based on the status change data and a defined recording format (S605) and recorded to the log file 411 (S606). Control then steps to S607. If the status change data is not data that requires recording (S604 returns no), control skips forward to step S607.

5 More specifically, in step S605, the status change data log entry is generated by converting the status change data bits to a more easily understandable text expression, and the time and date when the status data was received is then added to the text expression to produce the log entry. The log entry is then stored to the specified log file (S606). It should be noted here that
10 the log file 411 is a file stored to a hard disk or other data storage device of the host computer 101. The name, storage address, and maximum size of this log file are set in the recording conditions. When the log file recording the status change data reaches the maximum size specified in the recording conditions, the oldest log entry is deleted so that the newest entry can be added, or the oldest entry is
15 overwritten by the newest log entry. An example of the status change recording log written to the log file 411 is shown in Fig. 8. In this example, the status change log is recorded in ascending order according to the date and time of the change in status.

Step S607 then determines if the status change data evaluation process
20 (step S604) has been applied to all status change data. If it has (S607 returns yes), control moves to step S608. If the recording evaluation step has not been applied to all status change data (S607 returns no), steps S604 to S606 repeat until all status change data has been evaluated. After all status change data has been evaluated, the received status data is stored to the status data memory 410
25 (S608), and the status change data recording process ends.

The assignee of the present application has also developed a software utility (described in Japanese Patent Application JP 2001-1858 and in commonly assigned, co-pending U.S. Application Ser. No. (not assigned), entitled "Device environment configuration system, device environment configuration method,
30 and data storage medium therefor") which, by displaying a device tree organized

by device class or connection port based on the OPOS device environment configuration stored in the registry, enables the POS system developer to visually understand how the devices in a POS terminal system are connected so that the device information can be easily edited and stored to the registry. Fig. 10A shows an example of a device map organized by device class showing how the various peripheral devices are connected, and Fig. 10B shows a display organized by device connection port.

The process for setting the recording conditions used in the status change data recording process is described next with reference to the flow chart in Fig. 7. If the device settings screen is selected by a particular operation when the device for which status change data is to be recorded is selected from the device map shown in Fig. 10A or 10B (step S701), a device settings screen for setting the recording conditions for a particular device is presented (step S702). An example of a device settings screen is shown in Fig. 9. Such a device settings screen enables the user to select the data to record (the data to be logged), the name of the log file to which the data is written, and the maximum size of the log file (step S703). The recording conditions written to recording conditions memory 412 are then generated based on the device, information, log file name, and maximum file size specified in step S703 (step S704). The recording conditions are then stored to the recording conditions memory 412 (step S705) and the process for setting the recording conditions ends. The recording conditions are stored, for example, in the OPOS device environment configuration settings written to the registry.

It should be noted that the process for setting the recording conditions using screens such as shown in Fig. 9 and Figs. 10A and 10B is described here by way of example only. Alternatively, for example, a software utility specifically for setting the recording conditions could be used, or the recording conditions could be added to the registry using a registry editor provided by the operating system (such as the regedit.exe registry editor utility provided by the Windows operating system).

If the programmable functions of the device status monitoring system described above are correlated to the steps of the flow charts shown in Fig. 5 and Fig. 6, the status data receiving step corresponds primarily to step S501 and the status change data recording step to S502. The status change data editing step
5 corresponds primarily to S604 and S605. The status change data detecting step corresponds primarily to S603; the status change data evaluating step to S604; and the status change data converting step to S605.

A practical device status monitoring method with broad general utility can also be achieved by implementing the programmable functions of the above
10 device status monitoring system in a programmed computer. Those skilled in the art will be able to develop program code to implement the method based on the present disclosure without undue experimentation. The program code may be implemented in software or firmware.

As will be known from the preceding description of the invention, all
15 changes in device status can be recorded by providing, in an OPOS service object for controlling a device connected to the host computer through the operating system of the host computer in a POS terminal system, a function for recording status change data, which is indicative of a change in device status, to a file or other recording unit. The application is thus able to use all information about
20 changes in the status of a device.

By referencing the information written to the status change data log, the POS terminal system user or application developer, for example, can therefore know about problems with a device even when the status display of the device is hidden or the device does not have a status display.

25 Furthermore, the status change data recorded to the log file can be used for remote POS system maintenance by sending the data when an error occurs or at a regular interval from the host computer of the POS terminal system to a host server managing the POS system.

Yet further, the POS terminal system user can, for example, identify frequently occurring problems by reviewing the log, thus anticipate problems that are likely to occur again, and as a result be prepared to handle the problem.

Although the present invention has been described in connection with the
5 preferred embodiments thereof with reference to the accompanying drawings, it is to be noted that various changes and modifications will be apparent to those skilled in the art. Such changes and modifications are to be understood as included within the scope of the present invention as defined by the appended claims, unless they depart therefrom.